# ESTELLE SPECIFICATION OF MIL-STD 188-220 DATALINK LAYER*
# INTEROPERABILITY STANDARD FOR DIGITAL MESSAGE TRANSFER DEVICE SUBSYSTEMS

Hao Li Paul D. Amer
University of Delaware
Newark, DE 19716

Samuel C. Chamberlain
Army Research Laboratory
Aberdeen Proving Ground, MD 21005

**Abstract**

This paper presents the results of a contract between ARL and the University of Delaware to develop a formal specification of the link layer of 188-220 using the ISO International Standard Formal Description Technique Estelle. This formal specification aims at discovering and resolving ambiguities in the original English document that would cause interpretation problems for implementors. The specification considers Type 1 connectionless (CL) operation of the link layer. It contains the complete set of command and response PDUs for the CL mode (UI, XID, URR, URNR, TEST). The paper discusses state diagrams and state transition tables needed for the Estelle specification. It also summarizes several ambiguities that were discovered in developing the Estelle specification.

## 1. Introduction

The military standard "Interoperability Standard for Digital Message Transfer Device Subsystems" (MIL-STD-188-220) [5,6] represents the army's focused efforts to digitize the battlefield. The army is hoping that by 2001, all systems will either use MIL-STD-188-220 or whatever it has evolved into under configuration control. To ensure that the standard is free from ambiguities that might cause problems for implementors, we used the Estelle language, an ISO International Standard formal specification technique (FDT), to formally specify the Type 1 connectionless operation mode of the link layer. Our efforts were based on the May 1993 draft, a version that has since been updated several times. In the process of developing the Estelle specification, we documented several ambiguities which may cause incompatibilities among different implementations. These ambiguities were reported to the group developing 188-220 and in some cases were accounted for in later versions.

*Support, in part, by the US Army Research Office Scientific Services Program administered by Batelle (DAAL03-91-C-0034), and the US Army CECOM, Ft. Monmouth (under ARO Contract DAAH04-94-G-0093).

Estelle is an ISO FDT designed for specifying computer communication protocols such as MIL-STD-188-220 that are based on the ISO Reference Model [3,4,7]. The network community has long recognized the importance of such practice of developing formal specifications based on the standards written in ambiguous natural languages such as English. While English specification are often easier to manipulate in the short term, a formal specification removes much of the ambiguity inherent in the English language. The Estelle specifications of several well-known protocols exist in the literature and have brought fruitful results in ensuring compatible implementations [1,2]. It is our hope that our Estelle formal specification will contribute to furthering the correctness of MIL-STD-188-220 or whatever it evolves into under configuration control by 2001 and help future implementors to produce compatible implementations.

This paper is organized as follows. Section 2 gives an overview of MIL-STD-188-220, focusing on the part that is most relevant to our specification. Section 3 presents the state diagrams and state transition tables that our Estelle specification of is based on. It includes a discussion of typical example problems and ambiguities that we have found in the process of developing the Estelle formal specification[1]. Section 4 briefly concludes the paper.

## 2. Overview of MIL-STD-188-220

### 2.1 General Architecture

The general architecture of MIL-STD-188-220 uses the ISO 7-layer reference model. The May 1993 standard notes that the transport layer and session layer are null and only provide a pass-through service[2].

[1] We emphasize that the discussions here apply to the May 1993 version and as such will require modification to be compatible with later versions.

[2] The more recent April 1995 version uses TCP/IP in these layers and does not address layers 5, 6, 7.

The standard specifies two types of link layer operations: type 1 operation is the mandatory connectionless operation mode (acknowledged and unacknowledged); type 2 operation is the optional connection-mode operation.

With respect to this domain: Link Layer, Connectionless Operation Mode, Figure 1 shows the relevant Estelle architecture for our formal specification. Figure 1 shows: an array of stations (containing the link layer and the network layer in the figure) are sharing a common physical channel. Between layers inside each of the stations, there are interaction points such as NL and LN. Layer interactions are communicated through these interaction points. For example, a data request issued by the network layer may go through the interaction points to the link layer below.

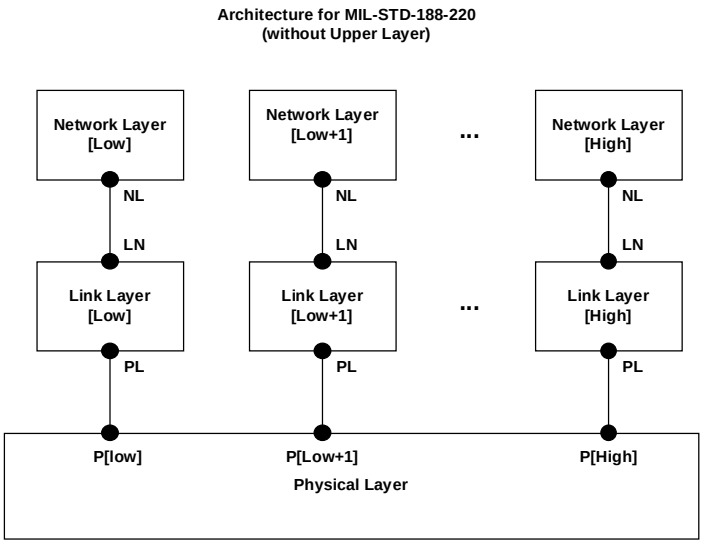


**Figure 1**

## 2.2 Net Access Control (NAC) Algorithm

It is clear from the above architecture that for multiple stations to share a common physical channel, a good multiple access control scheme is required. The one specified in the standard is essentially a variation of CSMA scheme with slotted acknowledgments, in which each station gets a slot to transmit its frame. Each station having a frame to send first listens to the channel and wait for its slot. If the channel becomes busy before the station reaches its slot, it shall withhold transmission and wait for the slot in the next round. Otherwise, the station may transmit its frame when its turn arrives.

The May 1993 draft specifies 3 schemes of assigning slots to stations: random (R-NAD), prioritized (P-NAD) and hybrid (H-NAD). For a detailed discussion of the various NAC algorithms, readers should refer to Appendix C of the MIL-STD-188-220.

## 3. Estelle Specification of the Link Layer, Connectionless Mode

Due to page limitations, we are unable to include the actual specification in this paper. We present here the state diagrams and state transition tables on which our Estelle formal specification is based. Readers who are interested in the full Estelle formal specification can request it from the authors.

### 3.1 State Diagrams and Transition Tables

To develop an Estelle formal specification of the standard, the first thing we need to do is to understand the inner workings of the standard and define finite state diagrams, because Estelle is based on communicating, extended finite state machines. This step is crucial for the actual formal specification. Once all states and transitions (including inputs and outputs) are finalized, the actual writing of Estelle is straightforward.

We divided the Connectionless Operation Mode of the Link Layer into two phases: the initialization phase in which a station upon physical set-up logically joins the net; and the operational phase in which a station logically in the net performs its normal operation (sending and receiving frames). Due to page constraints, only the operational phase is discussed here.

Figure 2 shows the state diagram for the operational phase. There are altogether 12 states and 26 transitions. Here we will just briefly explain the state transitions for sending out a frame in situations where the DL-DataReq arrives before the station reaches its slot and no other stations are transmitting before the station reaches its slot. For a full understanding of the state diagram, readers should refer to Table 3 (Transition Listing for the Operational Phase) and Table 4 (Transition Table for the Operational Phase). It is important to state that this transition table represents the authors' interpretation of the May 93 document. The authors expect others may have differing interpretation. This is natural in designing a protocol. The authors argue that design discussion and changes should be based on an unambiguous formal specification, not an inherently ambiguous English specification.

The link layer of a station logically in the net starts with the ACTIVE_IDLE state. Then it goes through transition 1 to the NAD_EXPIRED state, which means that the station has reached its slot, because the input to transition 1 is just "delay NAD". In the state NAD_EXPIRED, the link layer checks its queue for a DL-DataReq. If there is one, it would mean that the DL-DataReq arrived before NAD was reached. The link layer should go through transition 3, sending out the frame and enter the ACTIVE_IDLE state, if the DL_DataReq does not require an acknowledgment. If the DL_DataReq requires an acknowledgment, the station shall go through transition 2, sending out the frame and enter the WAITING_FOR_ACK state. Here readers might have noticed that because of a slight inconvenience in implementing the Quiet Timer in Estelle, we have introduced new states to keep the original semantics. After receiving an acknowledgment in the form of a URR PDU, the link layer goes through transition 11 and return to ACTIVE_IDLE state.

This is just one of many scenarios that can happen in the infinite range of possible procedures for a station to send out a frame. The complexity of the whole state diagram is not only because of these different scenarios and their variations, but also due to the fact that multiple stations need to access the same channel and the fact that the station also needs to listen to the channel and receive incoming frames.

### 3.2 Summary of Problems and Ambiguities

One goal in developing an Estelle formal specification was to discover and document problems and ambiguities that are commonly seen in a standard written in natural language. In the process of developing the Estelle specification, we documented more than twenty problems and ambiguities in the original English document. Here we will present one typical example of such findings. Readers who are interested in the full set of such problems and ambiguities can contact the authors.

The following two statements are taken from the original English document:

> "5.3.6.1.5.1 (p. 42) Sending UI command PDUs. Information transfer from an initiating station to a responding station shall be accomplished by sending the UI command PDU. When a sending station sends a UI command PDU with the P-bit set to 1, it shall start an acknowledgment timer for that transmission and *increment* an internal transmission count variable. If no URR response PDU is received before the timer runs out, the sending station shall resend the UI command PDU, *increment* the internal transmission count variable, and restart the acknowledgment timer. If a URR response PDU is still not received, this resending procedure shall be *repeated until* the value of the internal transmission count variable is equal to the value of the logical link parameter N4, as described in 5.3.7.1.1c, at which time an acknowledgment failure status shall be reported to the data-link user. An internal transmission count shall be maintained for each UI information exchange (where P-bit = 1) between a pair of sending and receiving stations."
>
> "c. (p. 54) Maximum number of transmissions, N4. N4 is a data-link parameter that indicates the maximum number of times that an UI or XID command PDU is sent by a station trying to accomplish a successful information exchange. Normally, N4 is set large enough to overcome the loss of a PDU due to link error conditions. The maximum number of times that a PDU is retransmitted following the expiration of the acknowledgment timer is established at protocol initialization. This value is in the range of 0 through 5 and defaults to 2."

These procedures have the potential problem of reaching an infinite loop. The counter is incremented twice (supposedly from 0 to 2) before reaching the "repeat until" clause in which it is incremented again and then gets compared with the N4 value. N4 is defined in the second paragraph of having the value 0 through 5 with 2 as the default value. If N4 has value 0, 1 or 2, the counter value will be greater than the N4 value before reaching the "until counter=N4" clause, thus resulting in an infinite loop.

Such problems and ambiguities are eliminated in a Estelle formal specification. Our specification makes the conditions for state transitions explicit through Estelle constructs. Indeed it was through the process of developing a formal specification that we were able to find such errors which are difficult to catch in normal reading.

## 4. Conclusion

In this paper, we have presented an Estelle specification of MIL-STD-188-220 (Link Layer, Connectionless Operation Mode, May 1993 version). We hope that this formal specification will help implementors resolve some of the ambiguities in the original English document (May 1993

version) that might hinder compatibility among different implementations. In the process of developing this Estelle formal specification, we have also discovered some problems and ambiguities in the original English document. The paper has described one typical example of such problems and ambiguities. The full Estelle formal specification and the full set of problems and ambiguities are available upon request. The authors are currently continuing their efforts by formally specifying the most recent MIL-STD-188-220 version (April 1995) for US Army CECOM.

## References


[1] P. Amer, F. Ceceli, "Estelle Formal Specification of ISO Virtual Terminal", *Computer Standards and Interfaces*, 9(2), Dec. 1989, 87-104.

[2] P. Amer, D. New, "Protocol Visualization in Estelle", *Computer Networks and ISDN Systems*, 25(7), Feb. 1993, 741-760.

[3] S. Budkowski, P. Dembinski, "An Introduction to Estelle: A Specification Language for Distributed Systems," *Computer Networks and ISDN Systems*, 14 (1), 1987, 3-24.

[4] Information Processing Systems -- Open Systems Interconnection: Estelle, A Formal Description Technique Based on an Extended State Transition Model, International Standard 9074, June 1989.

[5] Military Standard -- Interoperability Standard for Digital Message Transfer Device Subsystems (MIL-STD-188-220), 7 May 1993.

[6] J. Siliato, J. Latham, "Combat Net Radio (CNR) Protocols: A Means for Battlefield Digitization", Tech Report, U.S. Army Communications-Electronics Command, 16 November 1993

[7] R. L. Tenney, "Tutorial on Estelle and Early Testing", Tech. Report 92-1, University of Mass-Boston, Boston MA, 1992.


Table 1: Transition Table for the Operational Phase

| Transition | Input | Output |
|---|---|---|
| 1 | delay(NAD) | |
| 2 | DL_UnitDataReq(Reliability) | PL_UnitDataReq(UI COMMAND, P=1) |
| 3 | DL_UnitDataReq(NoReliability) | PL_UnitDataReq(UI COMMAND, P=0) |
| 4 | absolute (priority low) | |
| 5 | delay(NadMax - Nad) | |
| 6 | DL_UnitDataReq(NoReliability) | |
| 7 | DL_UnitDataReq(Reliability) | |
| 8 | absolute (priority low) | |
| 9 | PL_UnitDataInd(URR RESPONSE) | |
| 10 | delay(TP) | PL_UnitDataReq(UI COMMAND, P=1) |
| 11 | PL_UnitDataInd(URR RESPONSE) | |
| 12 | Counter = N4 | DL_StatusInd(ACK_FAIL) |
| 13 | PL_StatusInd(NET_BUSY) | |
| 14 | PL_UnitDataInd(UI, P=0) | DL_UnitDataInd, if address is in dest |
| 15 | PL_UnitDataInd(UI, P=1) | DL_UnitDataInd, if address is in dest |
| 16 | delay(TP) | |
| 17 | delay(RHD) (if address is in) | PL_UnitDataReq(Ack) |
| 18 | delay(TP - RHD) | |
| 19 | PL_UnitDataInd(URNR RESPONSE) | |
| 20 | PL_UnitDataInd(URR COMMAND) | |
| 21 | delay(NAD) | PL_UnitDataReq(UI COMMAND, P=1) |
| 22 | delay(NAD) | PL_UnitDataReq(UI COMMAND, P=0) |
| 23 | PL_StatusInd(NET_BUSY) | Set BacklogReliability flag |
| 24 | PL_StatusInd(NET_BUSY) | Set BacklogNoReliability flag |
| 25 | BacklogReliability = true | Clear BacklogReliability flag |
| 26 | BacklogNoReliability = true | Clear BacklogNoReliability flag |

Table 2: Transition Listing for the Operational Phase

| transition | explanation |
|---|---|
| | |
| 1 | The station reaches its slot after delaying R-NAD. |
| 2 | The station sends out a UI PDU (P=1) which was passed down to the link layer from the layer above before the slot is reached. |
| 3 | The station sends out a UI PDU (P=0) which was passed down to the link layer from the layer above before the slot is reached. |
| 4 | The station does not have a DL-DataReq from the layer above before reaching its slot. |
| 5 | The station has reached the biggest possible slot in the net. |
| 6 | A DL-DataReq has been passed down to the link layer in [NAD, 3/4NS] (P=0). |
| 7 | A DL-DataReq has been passed down to the link layer in [NAD, 3/4NS] (P=1). |
| 8 | The station does not have a DL-DataReq from the layer above before reaching the biggest slot on the net (3/4NS). |
| 9 | The sending station receives an ack from one of the addressed stations. It waits for the rest of the acks to come. |
| 10 | The acknowledgment timer expires. The sending station updates the destination addresses in the UI PDU and retransmits it. |
| 11 | The sending station has received acks from all the addressed stations. |
| 12 | The sending station has tried retransmission of the UI PDU N4 times, yet not all acks have been received. It reports failure to the layer above. |
| 13 | The net becomes busy before the station reaches its slot. |
| 14 | The incoming PDU has P=0. No ack period needs to be scheduled. |
| 15 | The incoming PDU has P=1. The station needs to schedule an ack period, the length of which depends on how many destination addresses are in the incoming PDU. |
| 16 | The station's address is not in the incoming PDU. The station only needs to wait for the ack period to expire. |
| 17 | The station's address is in the incoming PDU. The station sends out a response in the form of a URR PDU when the appropriate slot is reached (depending on the position the station's address appears in the group of destination addresses in the incoming PDU). |
| 18 | After sending out its own response, the station schedules an ack period for the rest of the addressed stations to send out an ack to the sending station. |
| 19 | While waiting for an ack, the station receives a URNR PDU from one of the addressed stations indicating busy condition. |
| 20 | After receiving a URNR PDU for some time, the station receives a URR from the same station which was experiencing busy condition before, indicating that the busy condition has been cleared. |
| 21 | The station sends out a UI PDU (P=1) of a DL-DataReq which arrives in [NAD, 3/4NS] upon reaching its new slot. |
| 22 | The station sends out a UI PDU (P=0) of a DL-DataReq which arrives in [NAD, 3/4NS] upon reaching its new slot. |
| 23 | The net becomes busy before the station reaches its new slot and sends out the UI PDU (P=1) of a DL-DataReq which arrives in [NAD, 3/4NS]. |
| 24 | The net becomes busy before the station reaches its new slot and sends out the UI PDU (P=0) of a DL-DataReq which arrives in [NAD, 3/4NS]. |
| 25 | The station has a backlogged UI PDU (P=1) to send because the interference from other stations' transmissions. |
| 26 | The station has a backlogged UI PDU (P=0) to send because the interference from other stations' transmissions. |

## Operational Phase

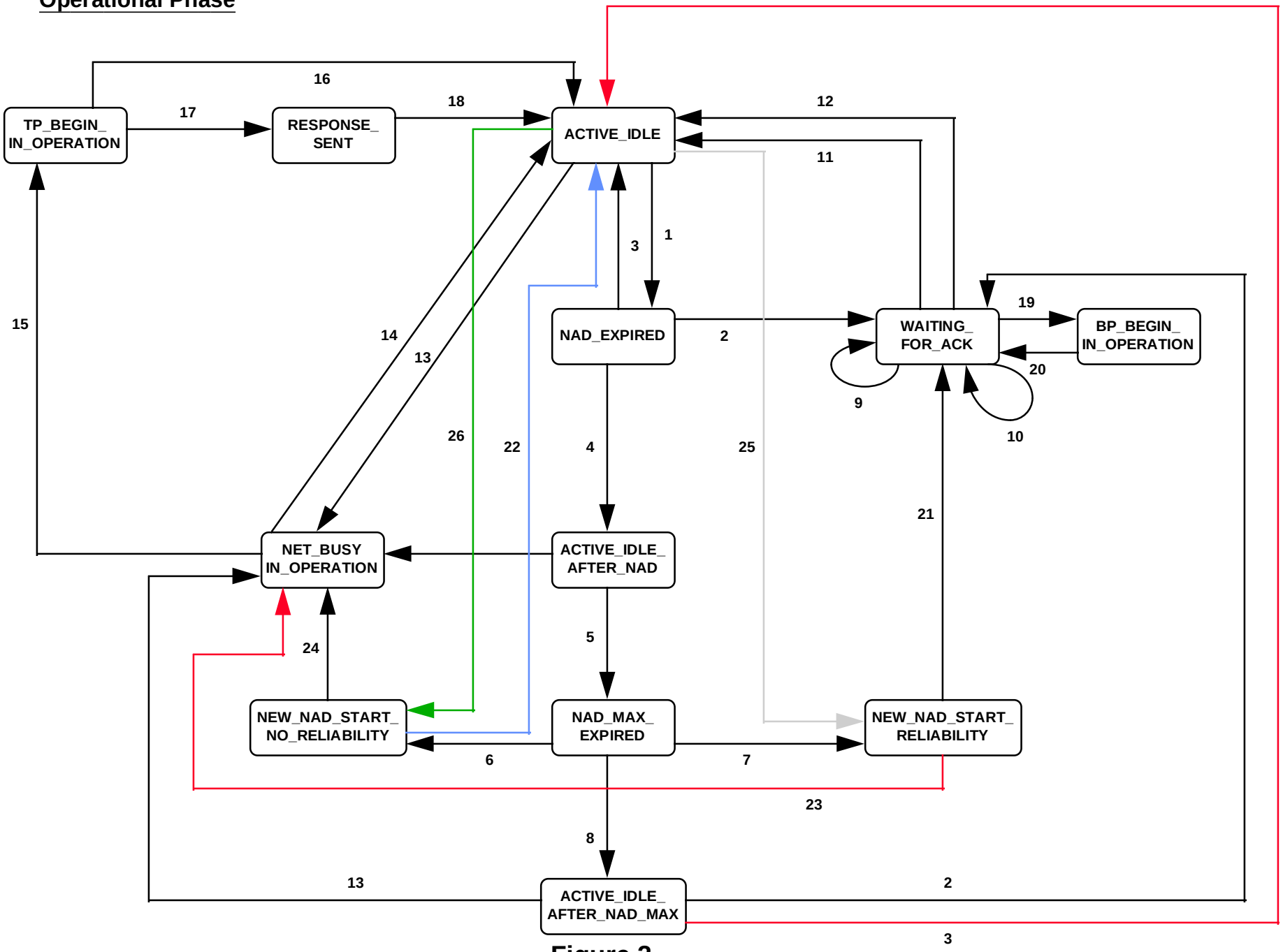


**Figure 2**